

# ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ «РАСЧЕТ-ЭМВ»

## Руководство по установке

### 1. Минимальный состав технических и программных средств

1.1. В состав технических средств должны входить:

1) для клиентской части ПО «Расчет-ЭМВ» (программный модуль RWEditor) ПЭВМ с характеристиками:

- универсальный процессор Intel Core I7 или аналогичный по производительности;
- оперативная память – не менее 16 Гб;
- HDD – не менее 1 Тб;
- видеокарта с производительностью не менее карты NVIDIA GTX 1080 (или аналогичной), поддерживающей OpenGL версии не ниже 3.3;
- монитор;
- мышь и клавиатура.

2) для серверной части ПО «Расчет-ЭМВ» (модули решатели) с учетом возможности массово-параллельных вычислений – вычислительный кластер в следующем техническом составе:

- не менее двух серверов с универсальными процессорами Intel Xeon 6 и выше, либо процессорами AMD EPYC и выше под управлением операционных систем CentOS 7.7 и выше, СПО Супер-ЭМВ 2.0 и выше, Арамид 3.3 и выше;
- оперативная память серверов – не менее 168 Гб;
- HDD сервера – не менее 1 Тб;
- видеоускоритель на сервере – NVIDIA V100 и выше;
- ПЭВМ для доступа к кластеру с вышеуказанными характеристиками.

1.2. В состав программных средств должны входить:

- операционная система Windows 10, Astra Linux 1.7, CentOS 7.7 и выше, СПО Супер-ЭМВ 2.0 и выше, Арамид 3.3 и выше;
- компиляторы GNU GCC версии 11.2.0-13.2.0 для серверной части ПО «Расчет-ЭМВ»;

– библиотеки окружения Microsoft Visual C++ Redistributable packages for Visual Studio 2019 (либо Microsoft Visual Studio 2019) для клиентской части ПО «Расчет-ЭМВ»;

– библиотека средств распараллеливания MPI – mvarich2 версии 2.3.6 для серверной части ПО «Расчет-ЭМВ»;

– набор библиотек инструментария разработки программного обеспечения для графических ускорителей NVIDIA HPC SDK 23.5 для серверной части ПО «Расчет-ЭМВ»;

– библиотека низкоуровневого распараллеливания для поддержки графических ускорителей nvptx-tools версии 1.0 для серверной части ПО «Расчет-ЭМВ»;

– фреймворк для разработки кроссплатформенных графических приложений Qt версии 5.15 для клиентской и серверной части ПО «Расчет-ЭМВ»;

– клиент удаленного доступа по протоколу SSH – openssh для клиентской и серверной части ПО «Расчет-ЭМВ».

Указанный состав библиотек входит в состав дистрибутива ПО «Расчет-ЭМВ».

## 2. Установка программы в среде ОС Linux

Для установки вспомогательного ПО необходимо выполнить установочный скрипт `thirdparty.sh`. В ходе выполнения этого скрипта будут собраны и установлены пакеты. Список пакетов и их версий приведен в таблице 1.

Таблица 1

Пакет	Версия
gcc	11.2.0
cmake	3.20.1
hdf5	1.10.5
mvarich2	2.3.6
qt	5.15.2
lib	1.0

Скрипт `thirdparty.sh` представляет собой самораспаковывающийся архив. Этот архив содержит каталог `thirdparty` соответственно со следующими каталогами и файлами:

- `install.sh`;
- `cmake`;
- `gcc`;
- `hdf5`;
- `mvarich`;
- `qt`;
- `lib`.

Файл `install.sh` осуществляет установку всех пакетов. После распаковки содержимого архива происходит выполнение этого файла.

Каталог `cmake` содержит файл `cmake-3.20.1.tar.gz` с исходными текстами пакета `cmake` и файл `inst`. Файл `inst` отвечает за сборку и установку пакета `cmake`.

Каталог `gcc` содержит файлы `binutils-2.38.tar.xz`, `gcc-11.2.0.tar.xz`, `gmp-6.2.1.tar.bz2`, `isl-0.25.tar.gz`, `mpc-1.2.1.tar.gz`, `mpfr-4.1.0.tar.gz`, `newlib-master.tar.gz` и `nvptx-tools-master.tar.gz` с исходными текстами, а также файлы `inst` и `inst.nvptx`. Файл `inst` отвечает за сборку и установку пакета `gcc`. Файл `inst.nvptx` отвечает за сборку и установку пакета `gcc` с поддержкой устройств `Nvidia`.

Каталог `hdf5` содержит файл `hdf5-1.10.5.tar.gz` с исходными текстами пакета `hdf5` и файл `inst`. Файл `inst` отвечает за сборку и установку пакета `hdf5`.

Каталог `mvarich` содержит файл `mvarich2-2.3.6.tar.gz` с исходными текстами пакета `mvarich` и файл `inst`. Файл `inst` отвечает за сборку и установку пакета `mvarich`.

Каталог `qt` содержит файл `qt-everywhere-src-5.15.2.tar.xz` с исходными текстами пакета `qt` и файл `inst`. Файл `inst` отвечает за сборку и установку пакета `qt`.

Каталог `lib` содержит набор опциональных для ПО «Расчет-ЭМВ» оптимизирующих библиотек.

Первым собирается компилятор `gcc` и окружение для его сборки. Далее с помощью этого компилятора собираются остальные пакеты. Возможна сборка компилятора с поддержкой ускорителей от компании `Nvidia`. В таблице 2 приведены пакеты, которые необходимо собрать для выполнения системных требований `gcc`.

Таблица 2

Пакет	Версия	Примечание
binutils	2.38	–
gmp	6.2.1	–
isl	0.25	–
mpc	1.2.1	–
mpfr	4.1.0	–
newlib	4.2.0	только для gcc с поддержкой nvidia
nvptx-tools	1.0	только для gcc с поддержкой nvidia

Пакеты, перечисленные в таблице 2, будут установлены в тот же каталог, что и компилятор gcc.

Сборка остальных, требуемых для системного окружения СПО пакетов, будет происходить в каталоге /tmp, при этом в конце сборки установочные файлы будут удалены.

Если существует необходимость установки дистрибутива СПО в конкретный каталог необходимо скрипту `thirdparty.sh` подать опцию `target`.

Например:

```
./thirdparty.sh --target /opt/thirdparty.
```

В ходе установки необходимо будет ответить на ряд вопросов:

```
- Input install dir [../lib]:
```

В этом случае запрашивается путь для установки вспомогательных библиотек, который может задаваться как относительным, так и как абсолютным. Значение по умолчанию: `../lib`.

Следующим вопросом определяется желаемое количество одновременно выполняемых нитей для сборки:

```
- Input max thread for build packages [16]:
```

В данном случае это параметр, который будет передан команде `make -j <значение>` для выполнения функции сборки всех пакетов. Значение по умолчанию: 16.

Следующим вопросом будет запрос на компиляцию под платформу GPU:

```
- Build gcc with nvidia support [Y/n]:
```

В данном случае определяется производить ли сборку компилятора gcc с поддержкой ускорителей nvidia. Значение по умолчанию: yes.

Для выполнения сборки СПО под GPU необходимо указать пути системного окружения:

```
- Input cuda dir [/opt/nvidia/hpc_sdk/Linux_x86_64/20.9/cuda/]:
```

Данной строкой задается системный путь до предварительно установленной библиотеки CUDA. На вычислительном узле или ПЭВМ должны быть установлены драйвера для ускорителей Nvidia, соответствующие этой версии библиотеки CUDA. Вопрос будет задан только в случае, если выбрана сборка gcc с поддержкой ускорителей Nvidia. Значение по умолчанию: /opt/nvidia/hpc\_sdk/Linux\_x86\_64/20.9/cuda/.

```
- Build gcc without nvidia support [Y/n]:
```

Следующим будет вопрос о необходимости сборки компилятора gcc. Этот вопрос будет задан, если на предыдущем шаге был выбран отказ от сборки gcc с поддержкой ускорителей nvidia. Значение по умолчанию: yes.

Далее идет вопрос о необходимости сборки пакета cmake. Значение по умолчанию: yes.

```
- Build cmake [Y/n]:
```

Далее идет вопрос о необходимости сборки пакета hdf5. Значение по умолчанию: yes.

```
- Build hdf5 [Y/n]:
```

Далее идет вопрос о необходимости сборки пакета mvapich. Значение по умолчанию: yes.

```
- Build mvapich [Y/n]:
```

Далее идет вопрос о необходимости сборки пакета Qt. Значение по умолчанию: yes.

```
- Build qt (need c++ compiler with c++11 standart) [Y/n]:
```

Далее идет вопрос о установке опциональных оптимизирующих библиотек. Значение по умолчанию: yes.

```
- Install extra lib [Y/n]:
```

Пример вывода установочного скрипта thirdparty.sh приведен на рисунке 1.

```

[ocorninayu@ccn1 ntrn]$ ./thirdparty.sh --target /home/o37/ocorninayu/src
Creating directory /home/o37/ocorninayu/src
Verifying archive integrity... 100% MD5 checksums are OK. All good.
Decrypting and uncompressing Thirdparty...
enter aes-256-cbc decryption password:
100%
Input install dir [../lib]: ../lib
Input max thread for build packages [16]: 8
Build gcc with nvidia support [Y/n]: n
Build gcc without nvidia support [Y/n]:y
===== GMP =====
===== ISL =====
===== MPFR =====
===== MPC =====
===== BINUTILS =====
===== GCC =====
---- GCC BUILD SUCCESS ----
Build cmake [Y/n]: y
===== CMAKE =====
---- CMAKE BUILD SUCCESS ----
Build hdf5 [Y/n]: y
===== HDF5 =====
---- HDF5 BUILD SUCCESS ----
Build mvapich [Y/n]: y
===== MVAPICH =====
---- MVAPICH BUILD SUCCESS ----
Build qt (need c++ compiler with c++11 standart) [Y/n]: y
===== QT5 =====
---- QT BUILD SUCCESS ----
Install extra lib [Y/n]: y
===== EXTRA LIB =====
---- EXTRA LIB INSTALL_SUCCESS ----

```

Рисунок 1

Сборка каждого пакета должна завершаться сообщением BUILD\_SUCCESS.

По умолчанию сборка всех пакетов происходит в каталоге /tmp, при этом в конце сборки установочные файлы будут удалены.

Чтобы задать каталог, в котором будет происходить сборка пакетов, необходимо скрипту thirdparty.sh подать опцию target.

Если при сборке пакета произошла ошибка – выдаётся сообщение BUILD\_FAIL. Причину ошибки можно посмотреть в каталоге, где происходила сборка соответствующего пакета. Для просмотра диагностических сообщений при сборке пакетов в каждом подкаталоге исходных текстов создаются файлы журналов с именами: logconf, logstrap, logmake, logmakeinstall.

В результате выполнения установочного скрипта должен быть создан каталог, имя которого было указано на первом шаге установки. Каждый пакет устанавливается в соответствующий подкаталог (cmake, gcc, hdf5, mvapich, qt).

Например, при выборе пользователем в качестве каталога установки вспомогательного ПО /home/user/empscore/lib в его домашнем каталоге будет создана следующая структура, приведенная на рисунке 2.

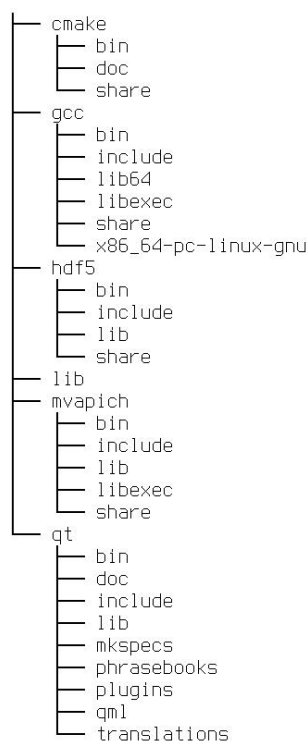


Рисунок 2

Установка дистрибутива ПО «Расчет-ЭМВ» осуществляется с помощью установочного скрипта `etalon.sh`.

В процессе установки будет запрошен пароль, сообщаемый разработчиком ПО конкретному заказчику.

Установка дистрибутива состоит из двух процедур:

- установка клиентской части (графическое приложение RWEEditor и набор решателей для ПЭВМ под управлением ОС Linux);
- установка серверной части (набор решателей и примеров скриптов для запуска ПО на вычислительной системе).

Диалоговая часть интерфейса установки показана на рисунке 3.

```

[ocorninayu@ccf1 Install]$ ./etalon.sh
Verifying archive integrity... 100% MD5 checksums are OK. All good.
Decrypting and uncompressing Etalon...
enter aes-256-cbc decryption password:
100%
Input install dir [/tmp/Etalon]:
Install user gui interface [Y/n]:
---- GUI INSTALL SUCCESS ----
Install cli solver [Y/n]:
---- CLI INSTALL SUCCESS ----

Install project done.

```

Рисунок 3

Полный состав дистрибутива приведен в таблице 3.

Таблица 3

Имя каталога	Описание
RWEditor	Содержит необходимые файлы для использования ПО «Расчет-ЭМВ» в графическом режиме (клиентская часть).
bin	Содержит набор решателей для использования в консольном режиме счета. Устанавливается на вычислительный кластер (серверная часть).
examples	Содержит набор примеров запуска расчета для системы управления кластером на базе ПО Slurm. Устанавливается на вычислительный кластер (серверная часть).

После завершения процедуры установки необходимо в окружении пользователя задать пути к библиотекам ранее установленного вспомогательного программного обеспечения. Например, пути можно определить в файле конфигурации командной оболочки BASH - `.bashrc` добавив туда строку следующего содержания:

```
export LD_LIBRARY_PATH=/home/user/empcore/lib:$LD_LIBRARY_PATH
```

В данной строке каталог `/home/user/empcore/lib` – это префикс-каталог, заданный в ходе установки вспомогательного программного обеспечения.

Кроме того, для взаимодействия клиентской части ПО «Расчет-ЭМВ» с лицензионным сервером требуется в системном окружении пользователя установить значения переменных окружения «LLIC\_HOST» и «LLIC\_PORT», в которых задается доменное имя лицензионного сервера и номер TCP-порта лицензионного сервера соответственно.

Лицензионный сервер устанавливается на ПЭВМ пользователя под управлением ОС Windows и входит в состав дистрибутива ПО «Расчет-ЭМВ» для ОС Microsoft Windows.



### 3. Установка программы в среде ОС Windows

Дистрибутив ПО «Расчет-ЭМВ» для ОС Microsoft Windows поставляется в виде RAR архива с именем Расчет-ЭМВ.rar. После распаковки архива структура каталога дистрибутива выглядит так, как показано на рисунке 4.

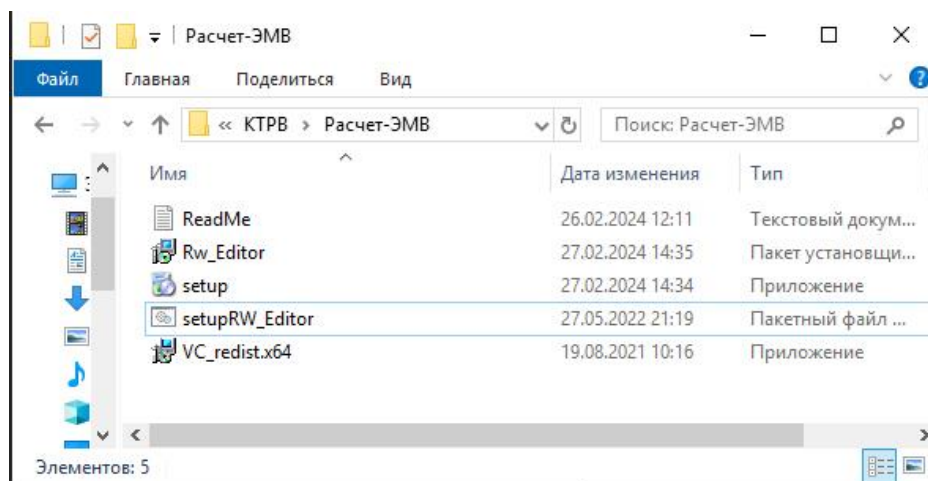
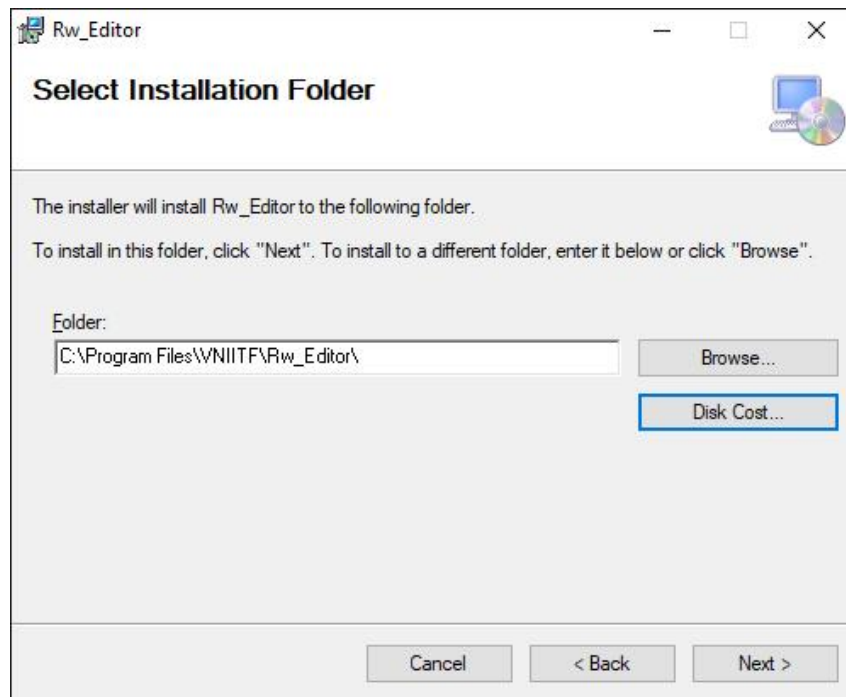


Рисунок 4

Для установки ПО «Расчет-ЭМВ» в среде ОС Windows необходимы права администратора операционной системы. От имени учетной записи администратора ПЭВМ необходимо запустить файл setupRw\_Editor.bat и следовать предлагаемым действиям программы установки. При установке предлагается путь по умолчанию C:\Program Files\VNITF\Rw\_Editor\. При необходимости можно установить свой путь, отредактировав его в строке ввода или выбрав конкретный путь, как показано на рисунке 5 и нажать кнопку «Далее».



Рисисунок 5

После установки ПО «Расчет-ЭМВ» автоматически запускается установка библиотек окружения Microsoft Visual C++ Redistributable packages for Visual Studio 2019 (либо Microsoft Visual Studio 2019) VC\_redist.x64.exe. В конце установки этого ПО необходимо поставить галочку о согласии.

Кроме того, в процедуру установки ПО «Расчет-ЭМВ» входит установка библиотек MPI: mpich2-1.4.1p1-win-x86-64.msi. Запуск установщика также выполняется автоматически после завершения установки библиотек окружения Microsoft Visual C++ Redistributable packages for Visual Studio 2019. При установке выполняется проверка на наличие ПО Microsoft .NET Framework 3.5, которое должно быть предварительно установлено в операционной системе.

В случае отсутствия установленного ПО Microsoft .NET Framework 3.5 оно может быть установлено путем выполнения следующей команды в оболочке PowerShell ОС Windows от имени администратора:

```
dism /online /enable-feature /featurename:NetFX3 /All /source:D:\sources\sxs /limitaccess
```

, где D:\sources\sxs – каталог дистрибутива операционной системы.

В случае, если необходимо провести установку обновлений ПО «Расчет-ЭМВ» права администратора не требуются. Для выполнения этой процедуры пользователь должен запустить полученный от разработчика файл

setup.exe, указать путь, куда была установлена предыдущая версия (например, C:\VNIITF\Rw\_Editor\)) и выполнить установку.

После успешной установки на рабочий стол ПЭВМ пользователя будут добавлены ярлыки для запуска программы RWEEditor, которая и будет содержать весь необходимый для использования ПО «Расчет-ЭМВ» в ОС Windows функционал.

В ходе установки ПО «Расчет-ЭМВ» также будет установлен лицензионный сервер.

Для настройки переменных окружения для взаимодействия клиентской части ПО «Расчет-ЭМВ» с лицензионным сервером требуется запустить с правами администратора утилиту control, предоставляющую доступ к классической панели управления ОС Windows, как показано на рисунках 6, 7.

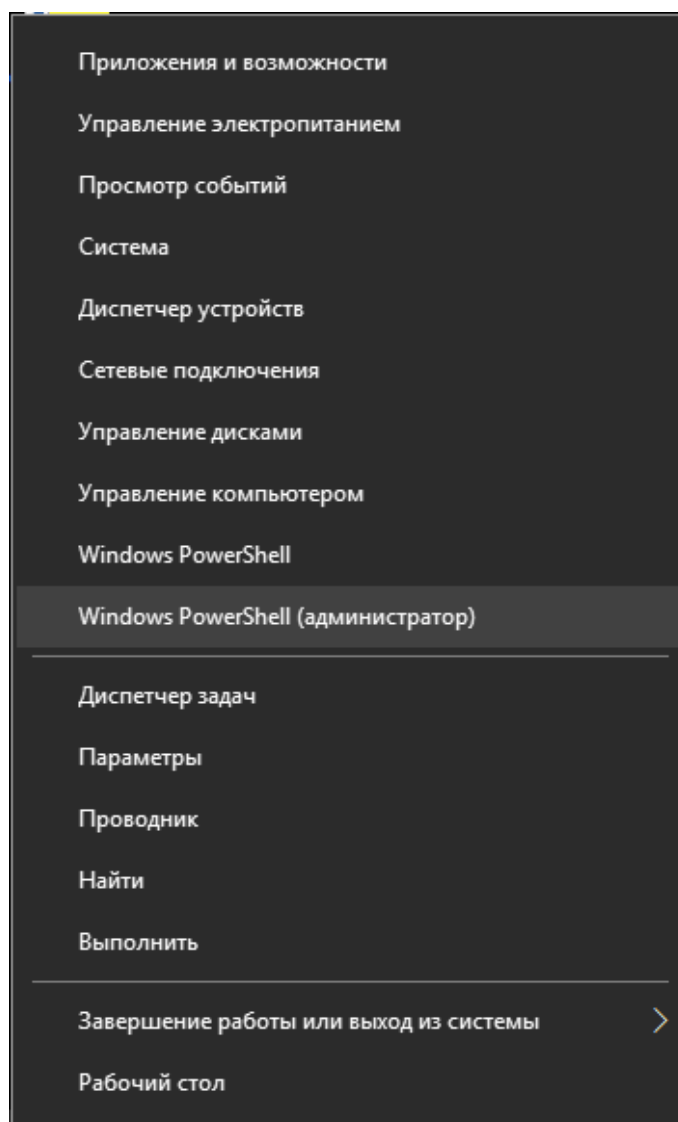


Рисунок 6

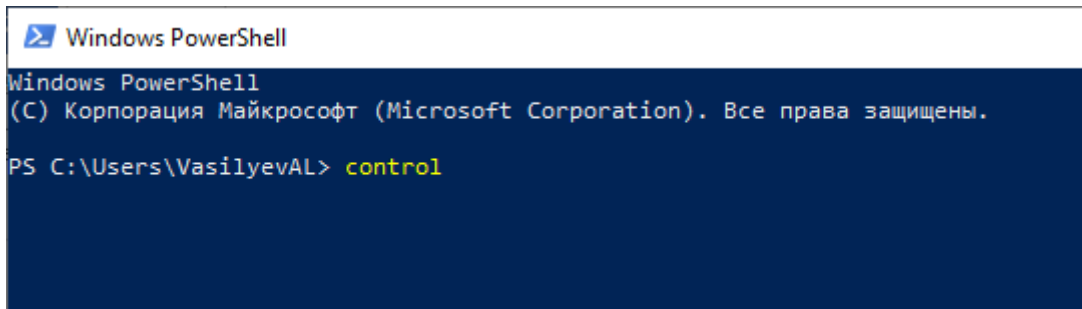


Рисунок 7

В появляющихся окнах последовательно выбрать разделы «Система и безопасность» -> «Система» -> «Дополнительные параметры системы», как показано на рисунках 8, 9, 10.

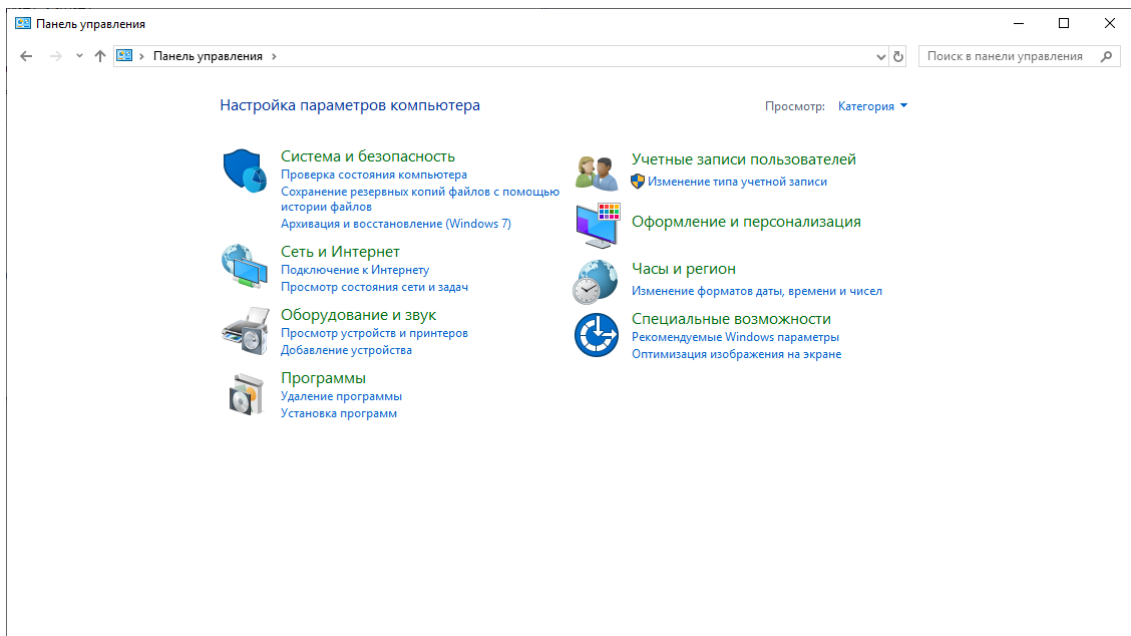


Рисунок 8

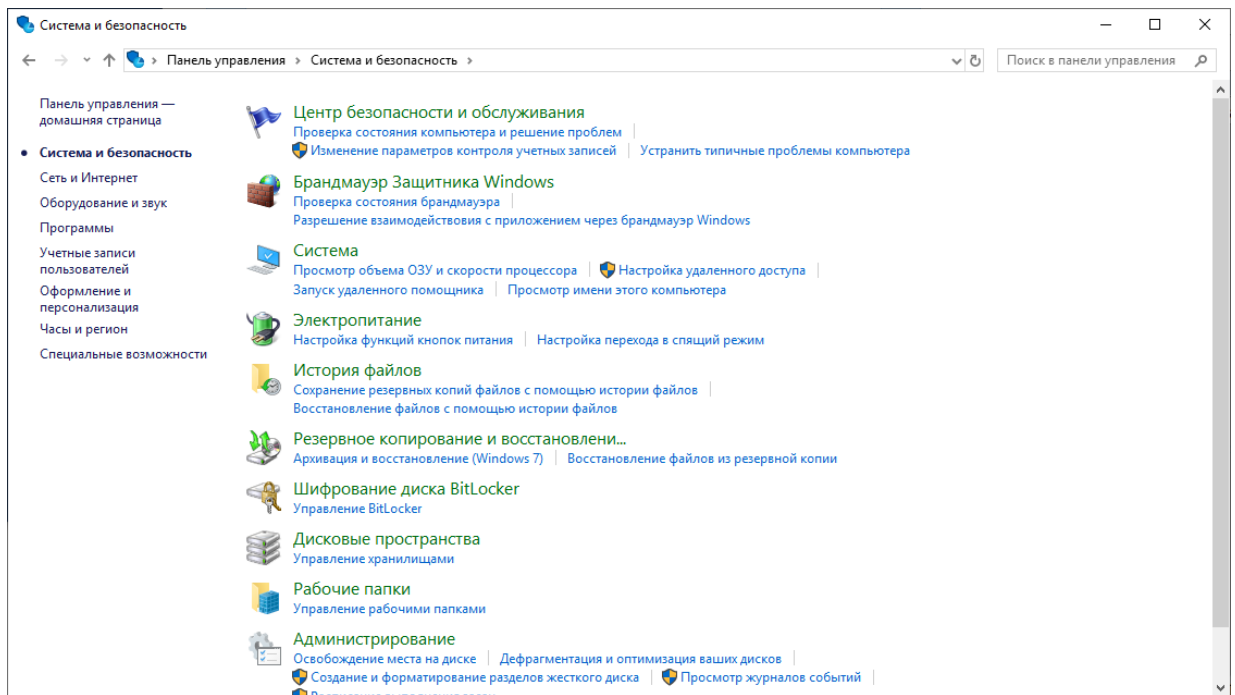


Рисунок 9

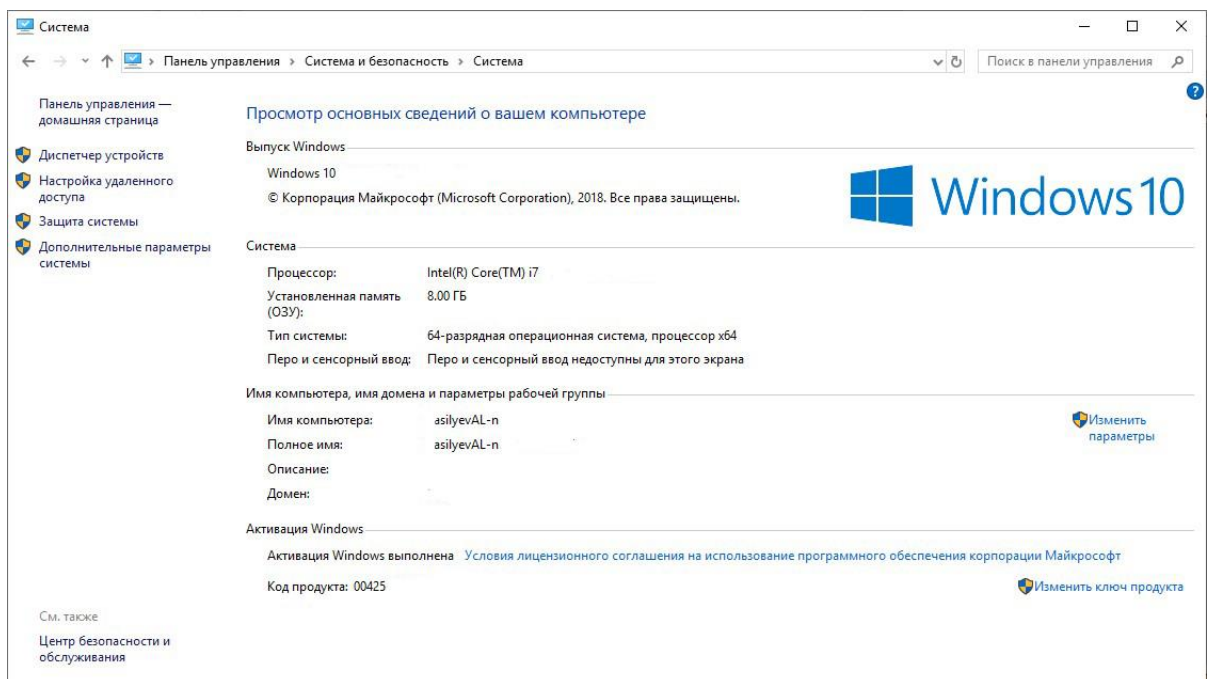


Рисунок 10

В окне свойств системы на вкладке «Дополнительно» нажать кнопку «Переменные среды», как показано на рисунке 11.

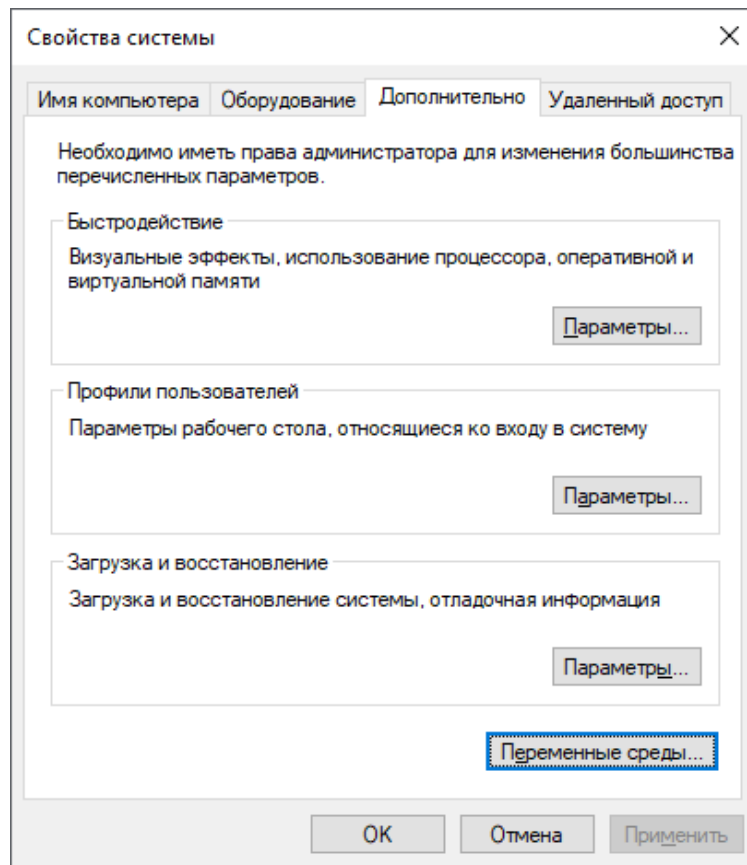


Рисунок 11

В окне переменных среды в разделе «Системные переменные» нажать кнопку «Создать», как показано на рисунке 12.

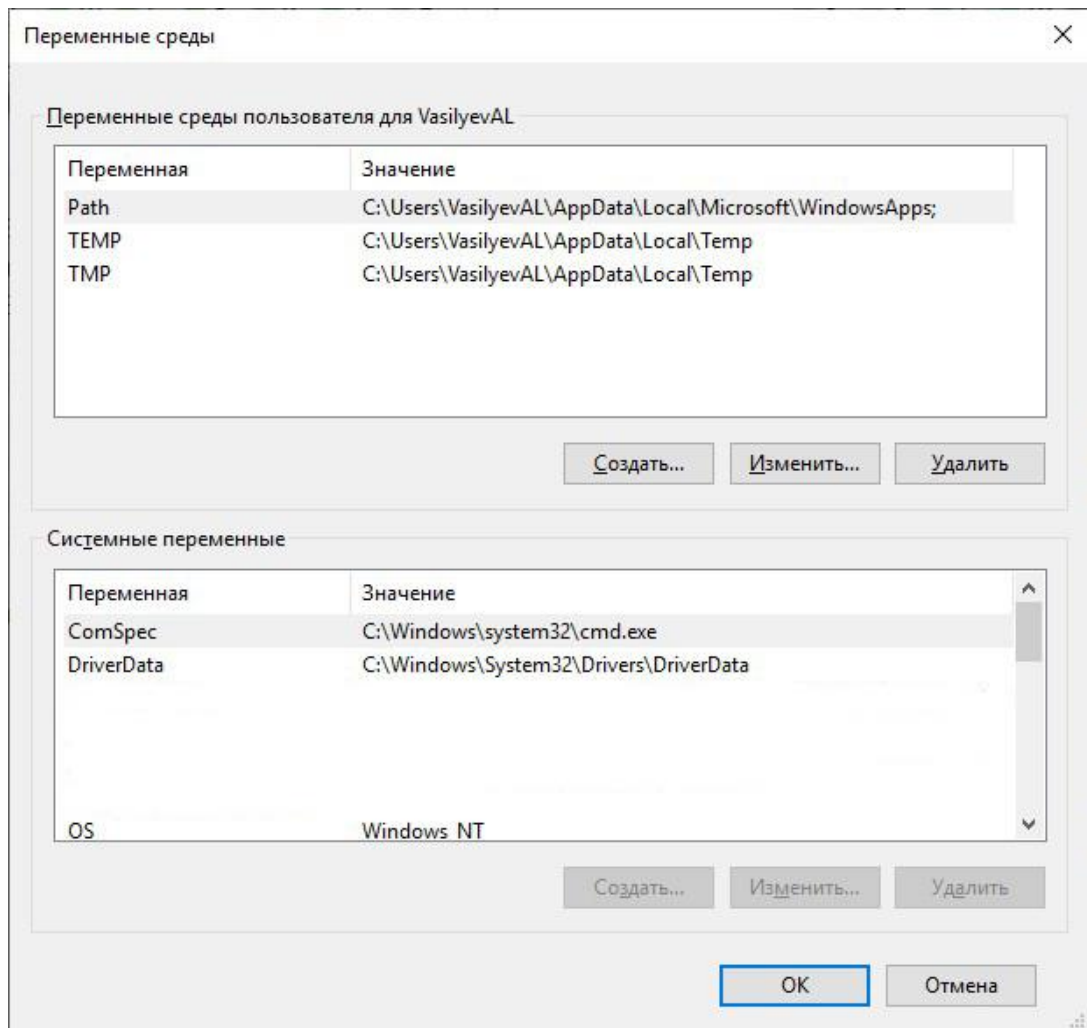


Рисунок 12

В окне создания новой системной переменной в поле «Имя переменной» прописать значение «LLIC\_HOST», в поле «Значение переменной» прописать IP-адрес лицензионного сервера, как показано на рисунке 13.

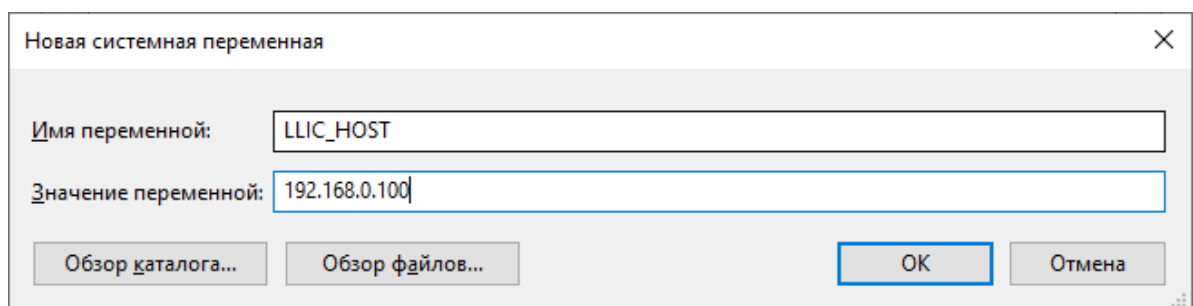


Рисунок 13

Аналогичным образом можно создать переменную «LLIC\_PORT», в которой задается номер TCP-порта лицензионного сервера, если в этом есть необходимость.

## ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

- ОС – операционная система
- ПО – программное обеспечение
- ПЭВМ – персональная электронно-вычислительная машина
- СПО – системное программное обеспечение
- GPU – graphics processing unit, графический процессор
- MPI – message passing interface, программный интерфейс для передачи сообщений между процессами одной задачи